Reconstruction Status

by Ryan Linehan, on behalf of
everyone working on reconstruction

Part 1: Beamline Reconstruction

So you have a bunch of sliced events. What can you do with them?

Answer: Produce reconstructed beamline objects!
Several objects (data products) exist in develop already:

1.) WCTrack

2.) TOF

3.) MuonRangeStackHits

4.) AGCounter

Each of these objects is produced by a module in the LArIATRecoModules directory:

1.) WCTrack: WCTrackBuilderSlicing_module.cc

2.) TOF: TimeOfFlightSlicing_module.cc

3.) MuonRangeStackHits: MuonRangeStackHitsSlicing_module.cc

4.) AGCounter: AerogelCherenkovCounterSlicing_module.cc

First Things First: Running the Reconstruction

Let's suppose you have a file of sliced events: SlicedEvents.root. To run only beamline reconstruction on it:

```
cd $MRB_TOP
lar -c srcs/lariatsoft/LArIATRecoModule/beamline_fullreco_lariat.fcl -s SlicedEvents.root
```

This produces WCTrack objects, TOF objects, MuonRangeStackHits, and AGCounter objects and puts them in the output file (default):

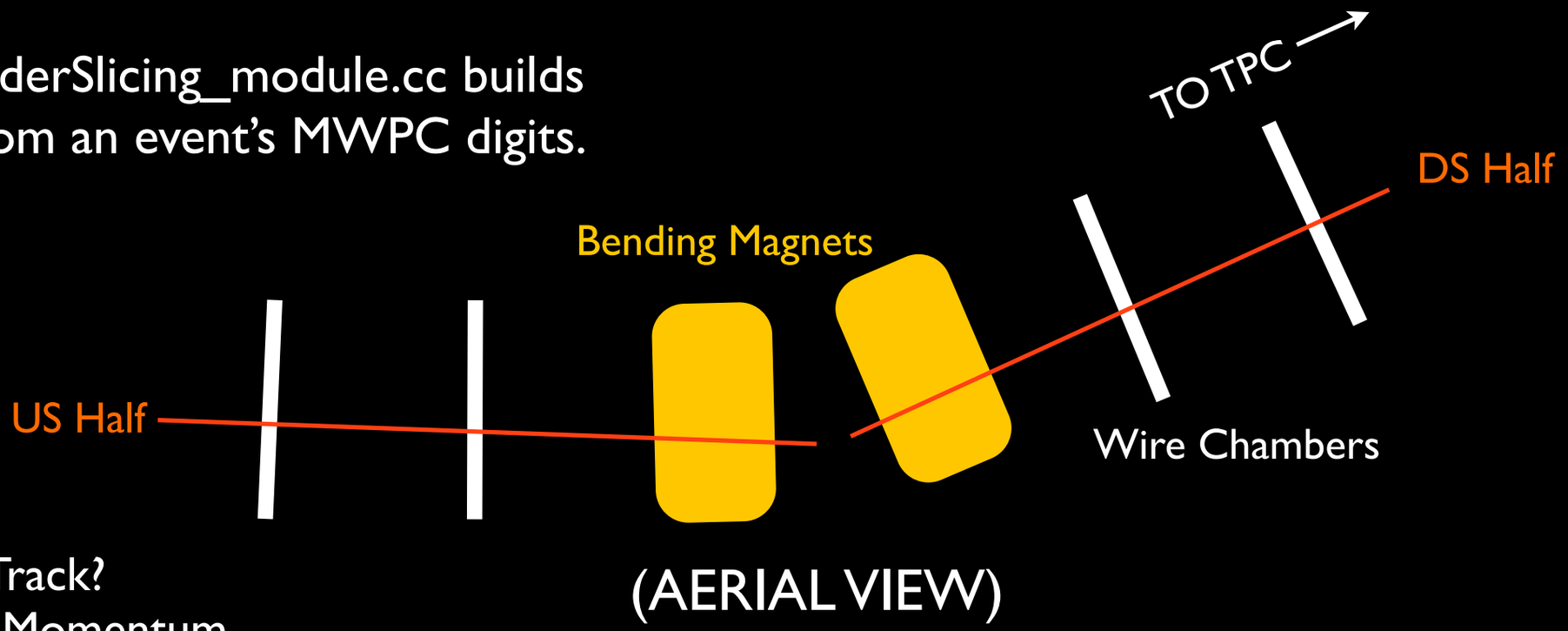
```
reconstructed_beamline.root
```

To see the parameter sets for the modules called in this file, look in:

```
srcs/lariatsoft/LArIATRecoModule/lariatbeamlinereco.fcl
```

WCTrack

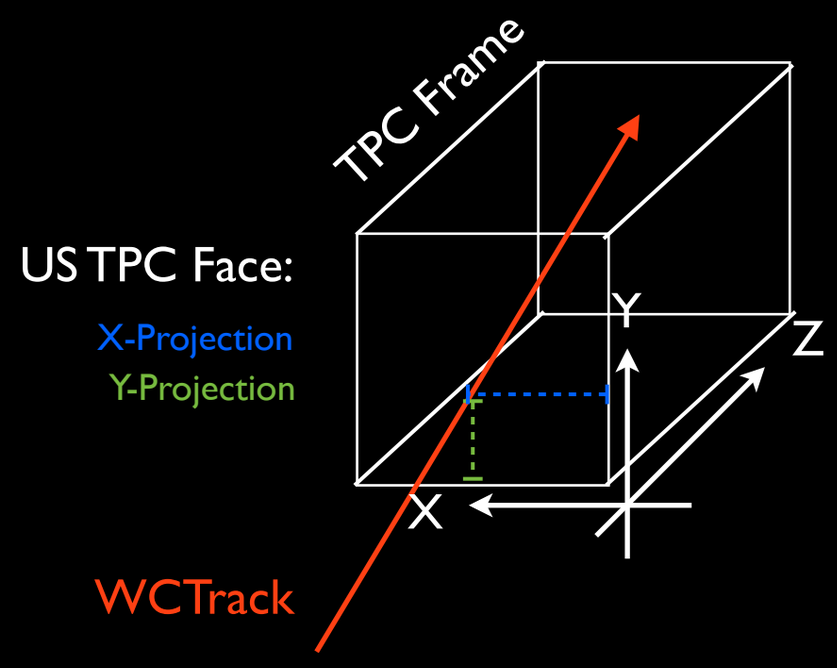
WCTrackBuilderSlicing_module.cc builds WCTracks from an event's MWPC digits.



What's in a WCTrack?
- Reconstructed Momentum
- Track quality information:
  + Vertical (Y) kink in US/DS halves
  + X,Y,Z Distance between US/DS half endpoints at bending magnets
- Track-On-TPC Info:
  + X/Y projections onto US TPC face
  + Theta in TPC coordinate system (polar angle)
  + Phi in TPC coordinate system (azimuthal angle)
- Hit information:
  + Hit WC #
  + Hit Wire
  + Hit Time

For specifics, look in:

`LArIATDataProducts/WCTrack.h`



TOF

TimeOfFlightSlicing_module.cc builds TOF objects from an event's time of flight counter digits.

What's in a TOF object?

- Number of times-of-flight: usually one.

- Vector of times-of-flight

- Vector of corresponding timestamps, where:

Timestamp = Time since spill started + DSTOF hit time

For more information on these variable names, look in:

`LArIATDataProducts/TOF.h`

MuonRangeStackHits

MuonRangeStackHitsSlicing_module.cc builds MuonRangeStackHits objects from an event's MuonRangeStack digits.

What's in a MuonRangeStackHits object?

- A map: paddle number to vector of hit times
  + Human readable form of digits' data

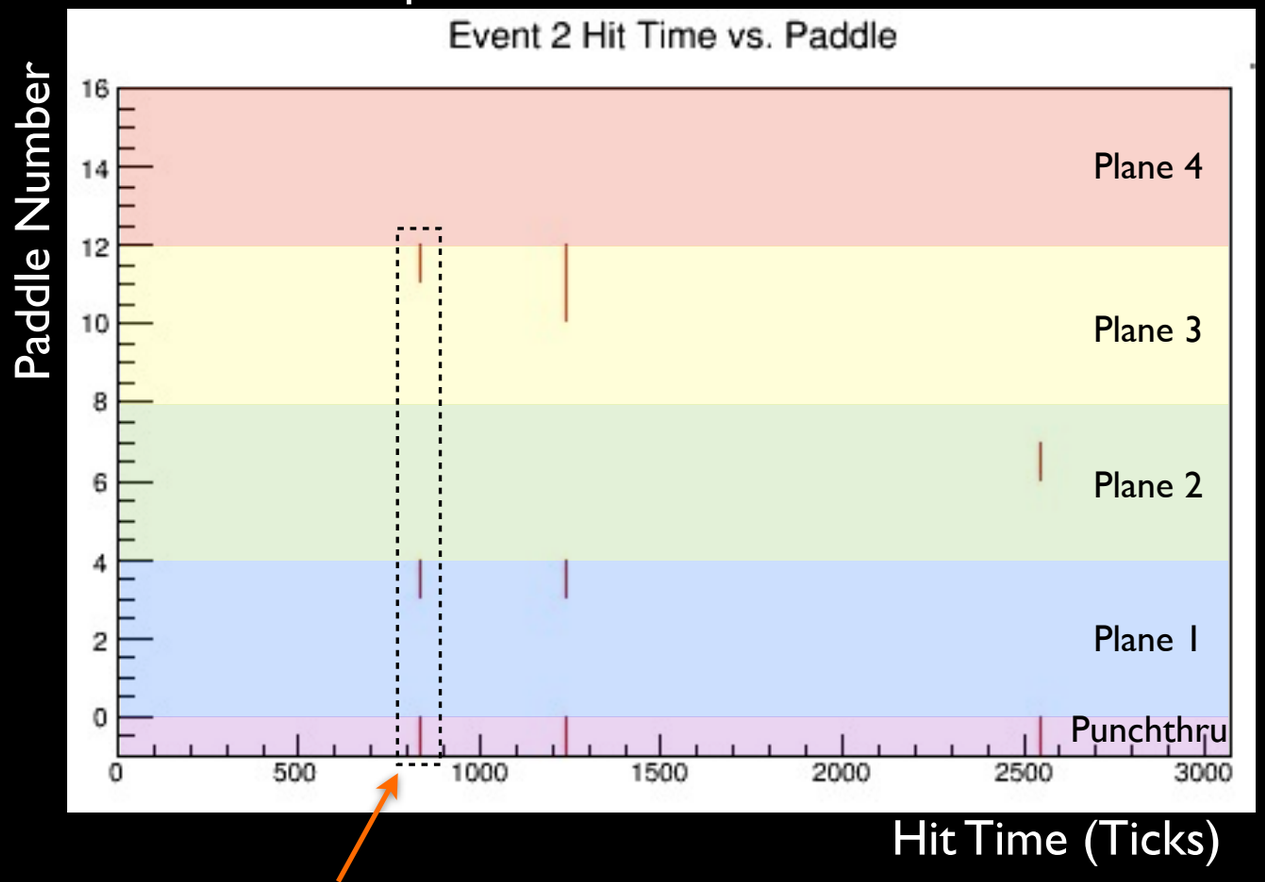
- A vector of MuRSTracks
  + MuRSTrack: built from coincidence of hits on 2 separate planes (including punchthrough)

- Info for each MuRSTrack:
  + Penetration depth
  + Arrival time (since hits are basically simultaneous)

For more specific information, look in:

`LArIATDataProducts/MuonRangeStackHits.h`

Example Hit Time vs. Paddle Plot



Example: hits grouped together as a MuRSTrack

Accessing these objects (in later modules)

Code snippet from an example analysis module

```
.
.
.

void ParticleIdentificationSlicing::produce(art::Event & e)
{
  //Get the collection of WCTracks produced by the WCTrackBuilder module
  art::Handle< std::vector<ldp::WCTrack> > WCTrackColHandle;
  e.getByLabel(fWCTrackModuleLabel,WCTrackColHandle);

  //Get the collection of TOF objects produced by the TOF module
  art::Handle< std::vector<ldp::TOF> > TOFColHandle;
  e.getByLabel(fTOFModuleLabel,TOFColHandle);

  //Get the collection of MuonRangeStackHits objects produced by the MuonRangeStackHitsBuilder module
  art::Handle< std::vector<ldp::MuonRangeStackHits> > MuRSColHandle;
  e.getByLabel(fMuRSModuleLabel,MuRSColHandle);
```

Use the getByLabel method to put the objects built by our beamline reco modules into handles...

```
  //Loop through the WCTracks and find the momentum for each
  for( size_t iWCTrack = 0; iWCTrack < WCTrackColHandle->size(); ++iWCTrack ){
    float reco_momentum = WCTrackColHandle->at(iWCTrack).Momentum();
  }

.
.
.
```

...then use the handles like pointers to vectors of the objects!

And that's it!

Beamline Reco: Future Plans

In the works:

- Particle ID based on WCTrack/TOF, MuRS, and Aerogel Counters is being worked on, and should be done relatively soon (~1.5 weeks)

- Track matching: WCTracks with TPC Tracks

Things for people to do:

- Testing and efficiency analysis of the WCTrack, TOF, MuRS, and Aerogel objects over many runs

- Once we've identified efficiencies, improvement/patching of existing "quick-and-dirty" algorithms for improved efficiencies:
  + WCTracking
  + TOF Finding
  + MuRS Tracking

Part 2: TPC Reconstruction

Updates: now using LArSoft v04_19_00 and lariatsoft v01_07_00

Run full reconstruction:

```
lar -c srcs/lariatsoft/JobConfigurations/Reco.fcl -s SlicedEvents.root
```

This does:

- Beamline reco:
  - +WCTrack
  - +TOF
  - +MuRS
- TPC reco:
  - + wire calibration
  - + hit finding
  - + cluster finding
  - + track finding

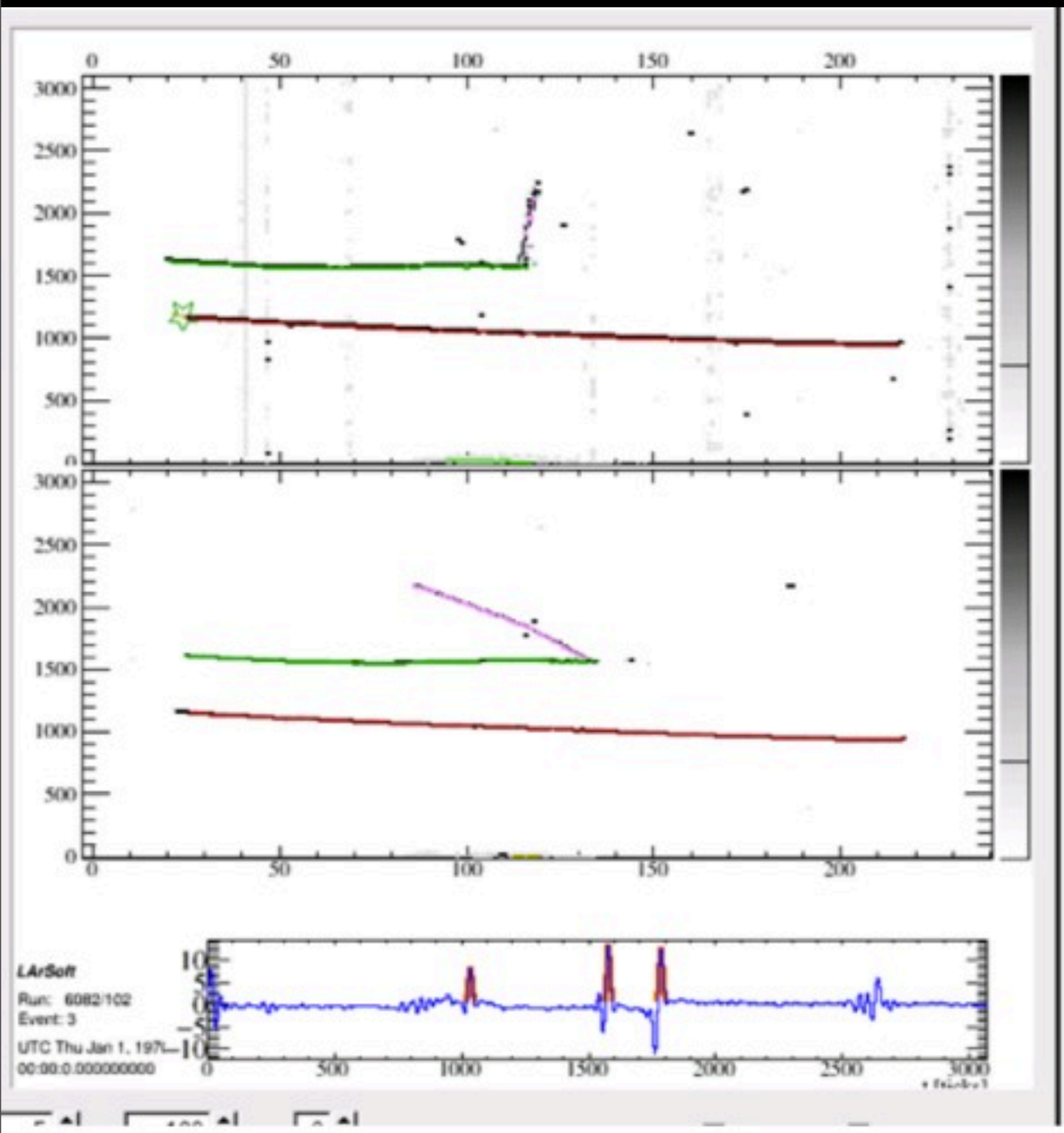
Updates to TPC Reco (from Tingjun)

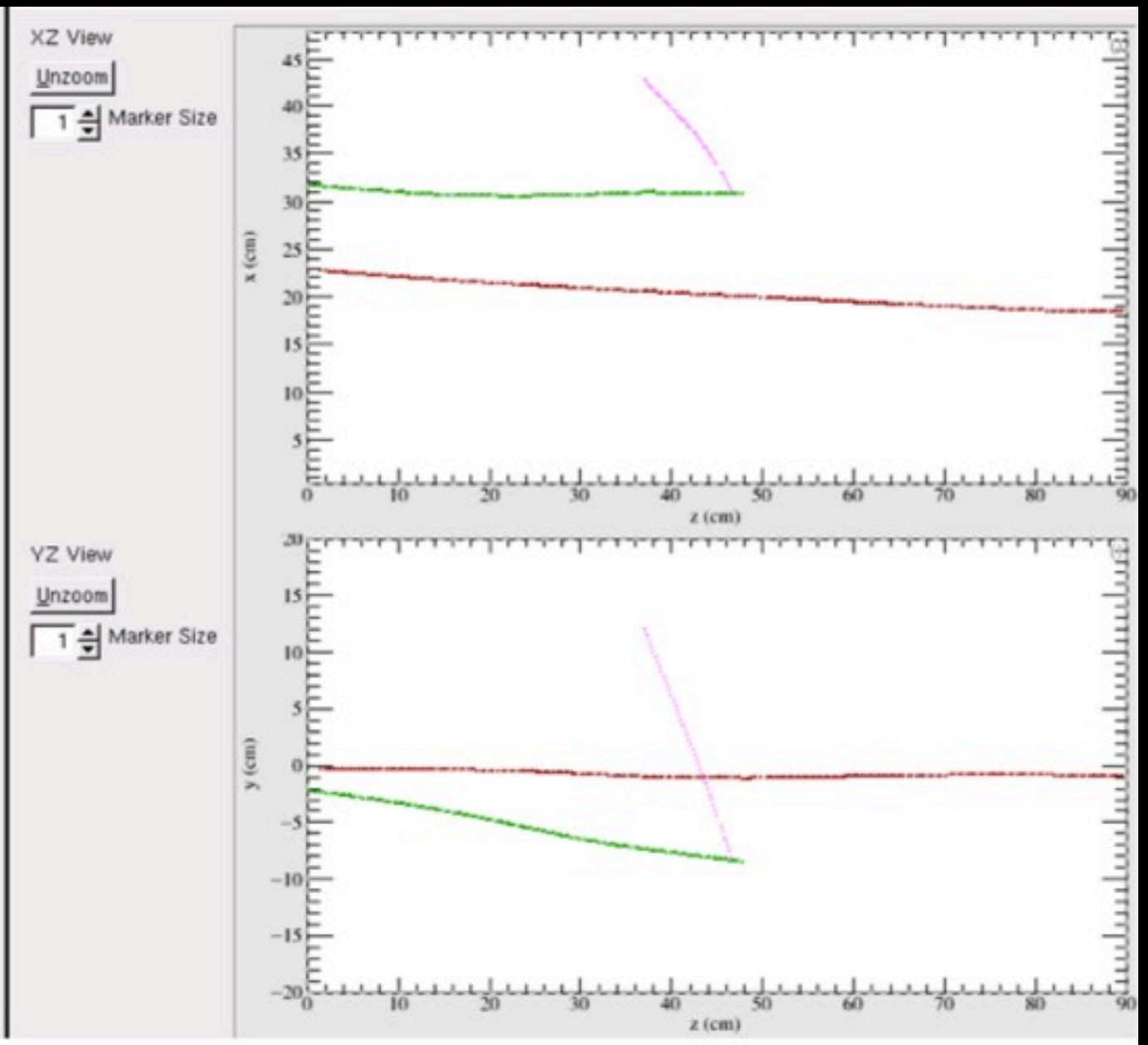
Fixed a time difference between U and V planes introduced by the field response function

- eliminated some accidental curvature in the tracks

Tuned cluster crawler

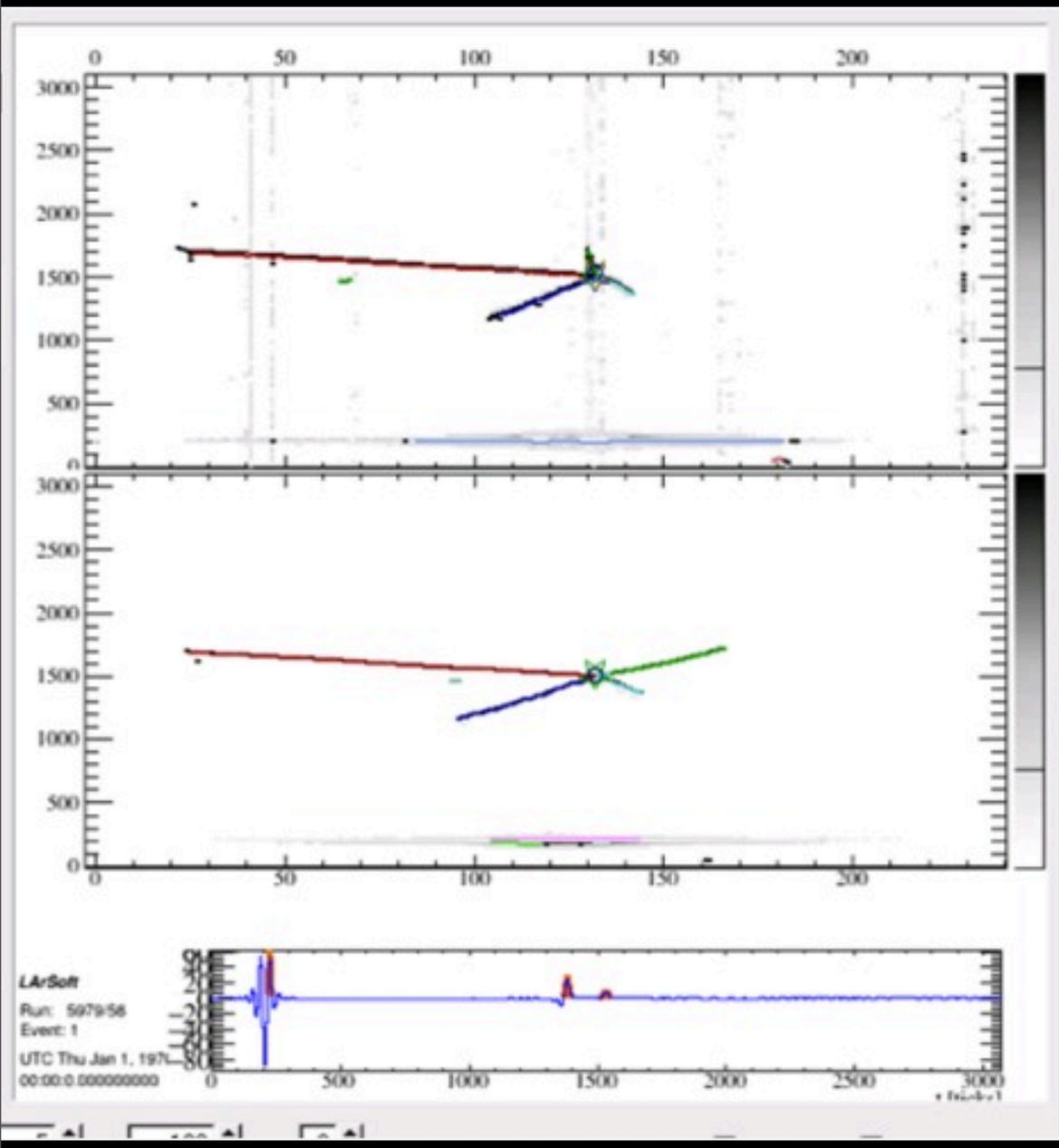
Event Examples with Tracking

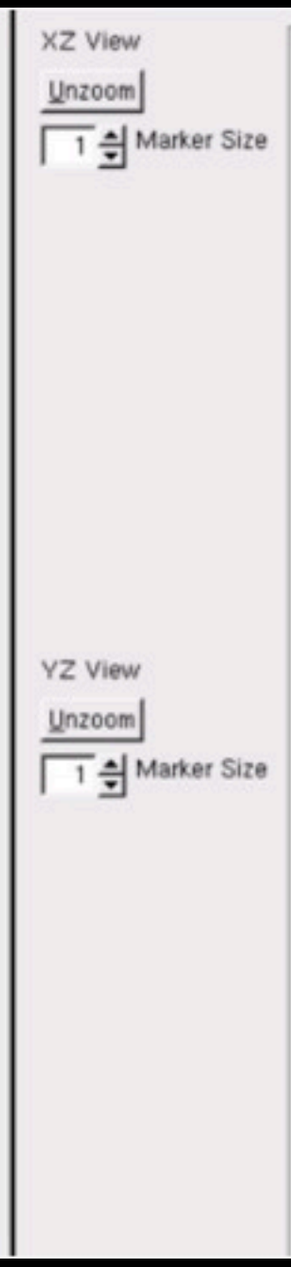


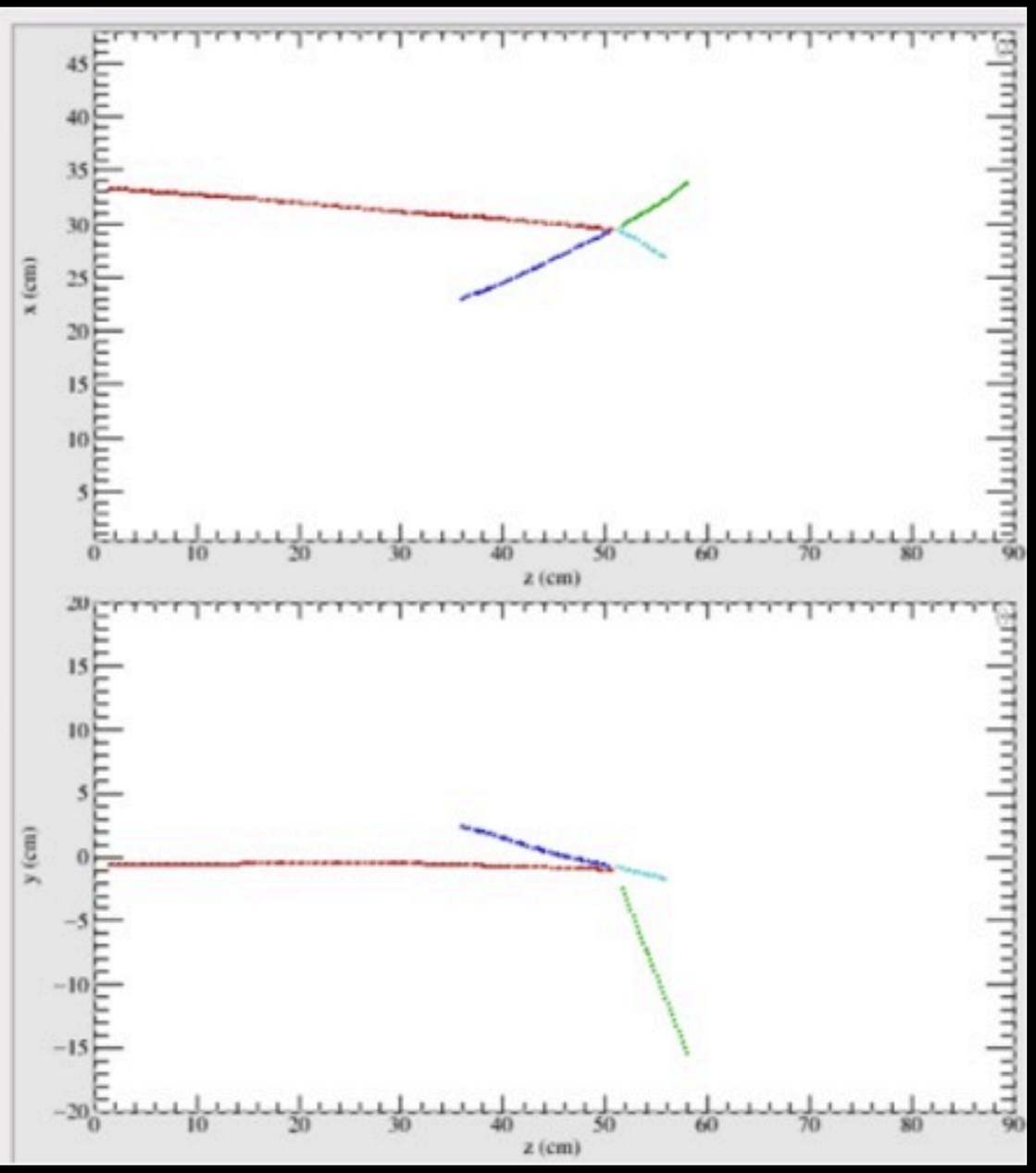


These tracks were produced with the **pmtrack** module - looks to be the best one available at the moment.

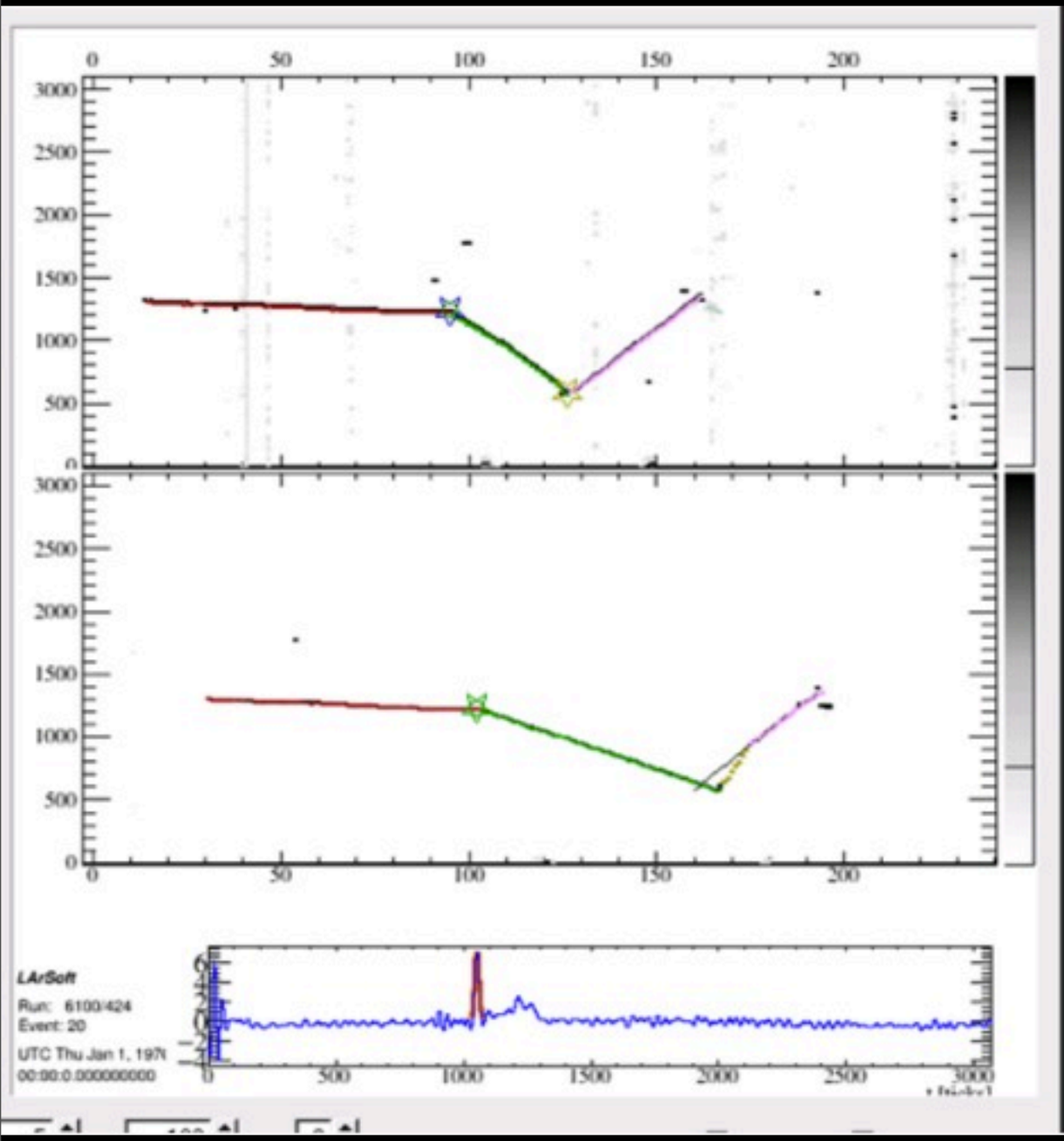
Event Examples with Tracking

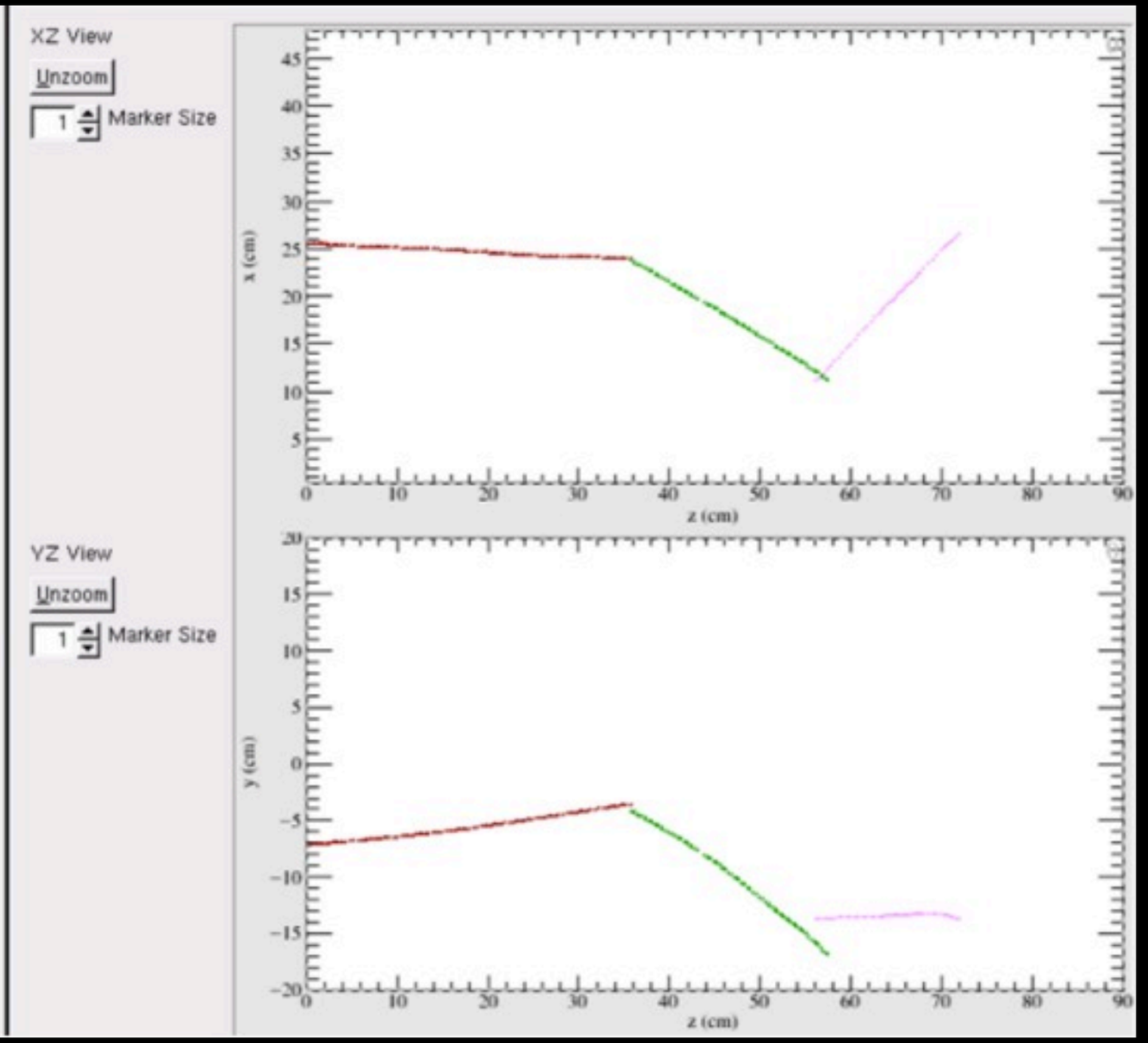






Event Examples with Tracking





Tracking still faces some small challenges, but on the whole, it looks really good and is ready to be used!

Summary

Beamline reconstruction and TPC reconstruction are moving quickly along.

We have 4 beamline reconstruction modules that produce these objects:

- WCTrack
- TOF
- MuonRangeStackHits
- AGCounter

We have a nearly complete TPC reconstruction in Reco.fcl with:

- Wire calibration
- Hit finding
- Cluster finding
- Track finding